# Instance-based Cost Sensitive Boosting

Ensieh Sharifinia and Reza Boostani

Faculty of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

**Abstract**

Since estimating the cost of each instance is an expensive and time consuming process, in the cost sensitive classification problems, experts empirically assign a constant cost value to all instances of each class. Since we know the importance of all samples is not equal, to overcome this drawback, a cost estimation approach is proposed here. We consider two separate costs for each instance, first the class cost and second the instance cost. The class cost can be given by expert or determined based on the ratio of population each class over all samples. The second cost is determined based on the probability of each instance within the estimated density of that class. The second contribution of this paper is to propose a Bayesian consistent ensemble learning scheme to deal with an estimated set of instance-based cost sensitive samples. In this way, the instance-based cost sensitive Bayesian decision rule is firstly derived. Secondly, an instance-based cost sensitive version of Bayesian consistent exponential loss function is proposed. Finally, upon the proposed loss function, the derivation of instance cost sensitive extensions of AdaBoost, RealBoost and GentleBoost are developed. When minimizing the instance-based conditional risk function, the resulted concave plot implies that each instance is classified according to its boundary. Experimental results on KDD98 and variety of UCI datasets demonstrate the superiority of the proposed algorithms compared to the conventional instance-based cost sensitive methods. Applying the pair of *T*-test to the results supports the significant supremacy of the proposed methods.

**Keywords:** *Instance-based cost sensitive, Bayesian consistent, Boosting, loss function.*

## 1. Introduction

Vast variety of applications such as medical diagnosis [1-5], fraud and intrusion detection [6-12] are naturally cost sensitive classification problems. We have the similar problem in the classification of imbalance datasets because the decision boundary is biased in favor of the class with larger population [13, 14]. In addition to consider this fact that the classes do not have the same cost of missing target, samples of each class do not have the equal importance or cost [15]. In practice, when gathering samples belong to a class, some samples carry noise in their measurement process, some others are naturally scattered from the center (so called outlier samples), some of them are near to the decision boarder (marginal samples) and others are far from it (confident or reliable samples). Considering this variety within the samples of each class in a real application, the assumption of assigning a constant cost value to all samples is very rough and therefore this viewpoint should be changed to avoid a significant drop in the performance of the cost-sensitive methods [16].

A lot of efforts have been made to determine a cost value to each instance, separately. One of the most important case is software cost estimation which is critical for the success of software project management [17,…]. Project cost estimation affect management activities such as resource allocation and project planning. Inaccurate cost estimation may causes serious problems to the company. As we see, in all cases the cost of instances are given by the problem and therefore the existing methods cannot be applied to the dataset where cost of instances are unknown. Since cost determination for these applications is really expensive and time consuming, researchers consider a constant cost for each class or simply use sampling methods to classify the samples. The first one is a rough approximation and the second one change the class distribution by adding interpolated samples.

Sampling methods and cost-sensitive learning algorithms are two basic approaches to handle imbalance datasets [18, 19]. Sometimes, over-sampling, under-sampling or their combination is used to equalize the population of all classes with the objective of enhancing the classification performance [20, 21]. For instance, discarding technique can be used to diminish the population size of bigger class while adding technique is used to resample or generate informative instances in those classes with low population. Nevertheless, finding a proper ratio for discarding or adding the examples is still one of the challenging issues in sampling techniques [22]. Of interest to this work is cost sensitive learning where the learner receives costs from user and decides in favor of cost minimization.

A binary classifier can be written as a mapping function $h(x): X \to Y$ which maps the input sample $X = \{x_1, x_2, \dots, x_n\} \epsilon \mathbb{R}^n$ to a class label $Y \epsilon \{1, -1\}$.

$$h(x) = sign[f(x)] \tag{1}$$

where $\text{sign(x)} = \begin{cases} 1 & if x \geq 0 \\ -1 & otherwise \end{cases}$ and $f(x): X \to \mathbb{R}$ is known as the classifier predictor. The following zero-one loss function has attracted the attention of researchers in the statistical machine learning field because each classifier that is designed based on the minimizing this function will be automatically a Bayesian consistent classifier [23].

$$L_{0/1}(h(x), y) = \begin{cases} 0 & if\ h(x) = y \\ 1 & if\ h(x) \neq y \end{cases} \tag{2}$$

Since in cost sensitive problems the misclassification costs for the positive and negative classes are different, the generalized form of the zero-one loss function for cost sensitive applications can be written as:

$$L_{C_1, C_{-1}}(\text{x}, \text{y}) = \begin{cases} 0, & \text{if } h(\text{x}) = \text{y}, \\ C_{-1}, & \text{if } h(\text{x}) = 1, \\ C_1, & \text{if } h(\text{x}) = -1. \end{cases} \tag{3}$$

In general, the Bayesian decision rule (BDR) can be applied to cost sensitive and cost insensitive classification problems and its results will be optimum for both situations [23, 24]. For the cost sensitive problem, BDR has the following form and the sign of function *f(x)* can determine the class label of each input sample *x*.

$$f(x) = log \frac{C_1 P(y=1\,|x)}{C_{-1} P(y=-1\,|x)} \quad (4)$$

The footprint of posterior probability ratio can be also seen in Neyman-Pearson lemma, similar to Eq. (1) [25]. The common problem with BDR and Neyman-Pearson lemma is to determine imprecise posterior probability when just low number of samples is available. The goal in detection problems is to find a classifier such that the risk function $E_{X,Y}[L(x,y)]$ is minimized. In practice, the cost of all inputs is not equal; consequently, each instance has its own cost and the generalized loss function is defined as:

$$L_{C_{X,Y}}(\mathrm{x},\mathrm{y}) = \begin{cases} 0, & \text{if } h(\mathrm{x}) = \mathrm{y}, \\ C_{X,Y}(\mathrm{x},-1), & \text{if } h(\mathrm{x}) = 1, \\ C_{X,Y}(\mathrm{x},1), & \text{if } h(\mathrm{x}) = -1. \end{cases} \quad (5)$$

Boosting algorithms have demonstrated good performance across a wide range of applications such as image retrieval [26, 27], multimodal speaker detection [28], and face recognition [29, 30]. Boosting algorithms are also employed in cost sensitive problems including medical diagnosis [31, 32, 33], fraud and intrusion detection [34, 35, 36], and computer vision [37, 38, 39, 40]. Although Masnadi-Shirazi and Vasconcelos [41] proposed a general framework to design cost sensitive loss functions to construct Boosting algorithms, their proposed methods could not cover the situation of considering a unique cost for each instance based on their importance.

A lot of efforts have been made toward incorporating instances' cost in boosting schemes by changing weight update mechanism or confidence function [42, 43, 44]. As well-known approaches in this area, three versions of cost sensitive boosting (CSBs) consider instances' cost in reweighting process when an instance is wrongly classified [43]. Moreover, there are some AdaBoost cost sensitive algorithms such as AdaC1, AdaC2 and AdaC3 that always consider costs in updating weights without considering whether the sample is wrongly or correctly classified [44]. The main flaw of the AdaCost, CSBs and AdaCs classifier families is the lack of statistical proofs to show that these two classifier families are Bayesian consistent. The lack of Bayesian consistency for a classifier would diminish the trust of researchers to employ it for their applications because there is no guarantee to rely on the optimality of the classifier.

The contribution of this study can be divided in two parts: first finding an automatic and reliable method to estimate the cost of each sample within its class. Second, develop an instance-based cost sensitive Bayesian consistent boosting algorithm in order to accurately classify the samples according to their cost values estimated in the first phase. The proposed method is applied to one instance-based cost sensitive dataset (KDD98) along with several datasets derived from the UCI datasets [45]. The results of the proposed method are compared to that of other mentioned instance-based cost sensitive methods including AdaCost, CSBs, AdaCs family classifiers.

The rest of this paper is organized as follows: In Section 2, the structure of boosting algorithm, the conventional cost based boosting algorithms and finally the proposed instance-

based cost sensitive methods are introduced. Section 3 is devoted to the experimental results and discussion. The paper is finally concluded in Section 4.

## 2. Methods and materials

In this section, first, the conventional boosting methods along with the proposed method are explained. Next, the employed datasets are expressed in terms of number of attributes and instances. Afterward, the evaluation criteria (e.g. metrics) and results representations are introduced.

### 2.1. Boosting framework

Schapire and Freund presented the idea of boosting in which tuning and combining of some weak learners can produce a strong learner. These learners are trained in a sequential manner and finally arranged in a parallel topology to take the final decision. They tried to prove the generalization of their algorithm by finding upper bounds on the empirical error and Vapnic-Chervonenkis dimension [46, 47, 48]. From another angle, Freidman *et al.* [49], statistically proved the generalization property of boosting framework and showed how loss function can affect the boosting performance. Let's denote $G_m(x)$ be the $m^{th}$ weak learner, and the decision maker of $f(x)$ is constructed by a linear combination of the trained weak learners, as described below [49]:

$$\mathrm{f(x)} = \sum_{m=1}^{T} G_m(x). \quad (6)$$

Where *T* is the number of weak learners and *f(.)* takes the final decision. Several versions of boosting algorithms such as AdaBoost, RealBoost [48, 49] and GentleBoost [49] have been introduced that all have the same topology but are different in their loss functions and optimization methods. What follows is a brief description of AdaBoost, RealBoost and GentleBoost.

#### 2.1.1. AdaBoost

AdaBoost is one of the most famous boosting algorithms which attain binary (discrete) weak learners $g_m$ and a multiplicand $\alpha_m$ as the confidence value of each weak learner. The standard Adaboost proposed by Schapire and Freund [48] was designed for a two class problem.

$$G_m^{Ada}(x) = \alpha_m g_m(x). \quad (7)$$

Where $G_m^{Ada}(x)$ is the weighted vote of the $m^{\text{th}}$ weak learner, $g_m(x)$ is the output of $m^{th}$ weak learner and $\alpha_m$is the coefficient of the $m^{th}$ weak learner. Freidman *et al.* [49] showed that by minimizing the expectation of exponential loss function ($E(e^{-yf(x)})$), the weak learner function and its coefficient are determined such that they try to minimize the total error of that round as described as follows:

$$g_m(x) = \arg\min_g(err_{(m)}) \quad (8)$$

$$err_{(m)} = \sum_{i=1}^{n} w_i^{(m)}[1 - I(y_i = g_m(x_i))] \quad \text{where} \quad I(x = y) = \begin{cases} 1, & if\ x = y, \\ 0, & if\ x \neq y. \end{cases} \quad (9)$$

Where $err_{(m)}$ is the total error of $g_m(x)$, and $I(.)$ is the indicator function. The confidence of the $m^{th}$ trained learner is determined as follows:

$$\alpha_m = \frac{1}{2}\log\frac{1-err_{(m)}}{err_{(m)}}, \quad (10)$$

As we see, the weights of misclassified samples (hard samples) are exponentially enforced while the weights of corrected classified samples are decreased by the following equation:

$$w_i^{(m+1)} = w_i^{(m)} e^{-y_i G_m^{Ada}(x_i)}. \quad (11)$$

After sequential training of binary learners, for each input sample, a weighted decision of different learners is determined and the decision is made based on a hard threshold (e.g. sign function) explained below:

$$f(x) = Sign(\sum_{i=1}^{T} \alpha_m g_m(x)) \quad (12)$$

Where *Sign(.)* is a famous hard threshold function and *f(.)* determines the label of each input sample for a two class problem. The final decision maker for an *M* class problem can be easily extended by taking an *argmax* on the votes of different weak learners and that label takes the majority vote is selected.

### 2.1.2. RealBoost

RealBoost is an extension of AdaBoost with a fairly similar reweighting scheme and the same decision function; however, its weak learners are not binary functions and able to produce real values. Hence, there is no need to estimate a confident parameter (learner's weight) for each learner to show its influence in the final decision process. In other words, real valued output does the same task as weighted binary classifier does in AdaBoost. To select a suitable classifier for RealBoost, gradient decent is applied to the exponential loss function leading to achieve the following weak learner:

$$G_m^{Real}(x) = \frac{1}{2}\log\frac{P_{Y|X}^{(w)}(1\,|\varphi_m(x))}{P_{Y|X}^{(w)}(-1\,|\varphi_m(x))} \quad (13)$$

Where $\varphi_m(x)$ is the feature response to *x* and index '*w*' indicates the estimated probability distribution after the reweighing process. The reweighting scheme of RealBoost is introduced below:

$$w_i^{m+1} = w_i^m \times e^{-y f_m^{real}(x)}. \quad (14)$$

Where *m* is the iteration number. The final decision of RealBoost for each input pattern *x* is determined Eq. (6) where $G_m^{Real}(x)$ is weak learner.

### 2.1.3. GentleBoost

GentleBoost is an extension of RealBoost which uses Newton method for minimizing the exponential loss function $E(e^{-yf(x)})$. The $m^{\text{th}}$ weak learner of the GentleBoost algorithm is described as follows:

$$G_m^{Gentle}(x) = P_{Y|X}^{(w)}(1\,|\varphi_m(x)) - P_{Y|X}^{(w)}(-1\,|\varphi_m(x)) \tag{15}$$

Where $\varphi_m(x)$ is the feature response to *x* and index '*w*' indicates the probability distribution after the reweighing process. The reweighting scheme of GentleBoost is the same as RealBoost; however, its weak learners are $G_m^{Gentle}(x)$ which are found by the Newton method are different to that of RealBoost. When an iteration of Newton algorithm is executed, a new weak learner is added to the current structure in order to produce a better estimator.

### 2.2. Cost sensitive classification

If we take a statistic from the published papers in cost sensitive learning, we can see that most of them simplify the task and consider a constant cost for each class. However, the applications are actually instance based cost sensitive problem. In order to address the instance cost sensitive classification task, it is essential to find the optimal BDR predictor in this case and change the selected loss function in a way that as the sample size grows, the function which is minimized the loss function converges to the optimal BDR predictor. Every classifier which tries to minimize this kind of loss function would be asymptotically optimal.

#### 2.2.1. Instance based cost sensitive BDR

Consider $L_{C_{X,Y}}(\mathrm{x},\mathrm{y})$, described in Eq. (5) which is the instance cost sensitive form of $L_{0,1}$ loss function. The optimal predictor is given by BDR,

$$f^*(x) = \log\frac{P_{Y|X}^{(c)}(1|x)}{P_{Y|X}^{(c)}(-1|x)} \tag{16}$$

where

$$P_{Y|X}^{(c)}(y'|x) = \frac{C_{X,Y}(\mathrm{x},\mathrm{y}')P_{Y|X}(\mathrm{y}'\,|\,\mathrm{x})}{\sum_{y\in\{-1,1\}} C_{X,Y}(\mathrm{x},\mathrm{y})P_{Y|X}(\mathrm{y}\,|\,\mathrm{x})} \tag{17}$$

The proof is brought in Appendix A which demonstrates that Eq. (16) is fully compatible with BDR in the sense that it reduces to the latter when costs are equal.

#### 2.2.2. Instance based cost sensitive exponential loss

The necessary condition of loss function for cost sensitive optimality is using the optimal cost sensitive predictor $f^*(\mathrm{x})$ of Eq. (16). To convert a simple boosting algorithm to an Instance-based Cost Sensitive (ICS) boosting, its loss function should be changed such that the new loss function asymptotically minimized by $f^*(\mathrm{x})$ which satisfy the optimality condition.

**Lemma 1.** Consider the following instance-based cost sensitive risk function:

$$R(x,y) = E_{X,Y}\left[I(y=1)e^{-C_{X,Y}(x,1)f(x)} + I(y=-1)e^{C_{X,Y}(x,-1)f(x)}\right]. \quad (18)$$

When the above expectation (risk function) is minimized by the asymmetric logistic transform of $P_{Y|X}(y|x)$, the following decision rule is estimated as the weak learner of an ICS boosting algorithm.

$$f(x) = \frac{1}{C_{X,Y}(x,-1)+C_{X,Y}(x,1)} log \frac{P^{(c)}_{Y|X}(1|x)}{P^{(c)}_{Y|X}(-1|x)} \quad (19)$$

The proof is brought in Appendix B. In order to investigate the effect of ICS exponential loss function on minimum conditional risk, let's consider the cost of false negative detection be one ($C_{X,Y}(x,-1)=1$) and call $C_1 = C_{X,Y}(x,1)$ and $\eta = P^{(c)}_{Y|X}(1|x)$ so, $P^{(c)}_{Y|X}(-1|x)$ would be $1-\eta$. Rewriting the conditional risk results to the following relation:

$$R(x,y) = E_{Y|X}\left[I(y=1)e^{-f(x)} + I(y=-1)e^{C_1 f(x)}\,|x\right] \quad (20)$$

And by replacing Eq. (19) as the best predictor, the minimum conditional risk would be estimated as follows:

$$R^*_L(C_1,\eta) = \eta\left(\frac{\eta}{1-\eta}\right)^{-\frac{C_1}{1+C_1}} + (1-\eta)\left(\frac{\eta}{1-\eta}\right)^{-\frac{1}{1+C_1}} \quad (21)$$

To see how the function $R^*_L(C_1,\eta)$ behaves over its variables space, the following figure is demonstrated (Fig. 1) where all the samples based on their costs and $P^{(c)}_{Y|X}(1|x)$ lies on the surface. Fig. 1 shows that the boundary tends toward negative class as the costs of instances increases which implicitly shows that each instance is classified according to its boundary.

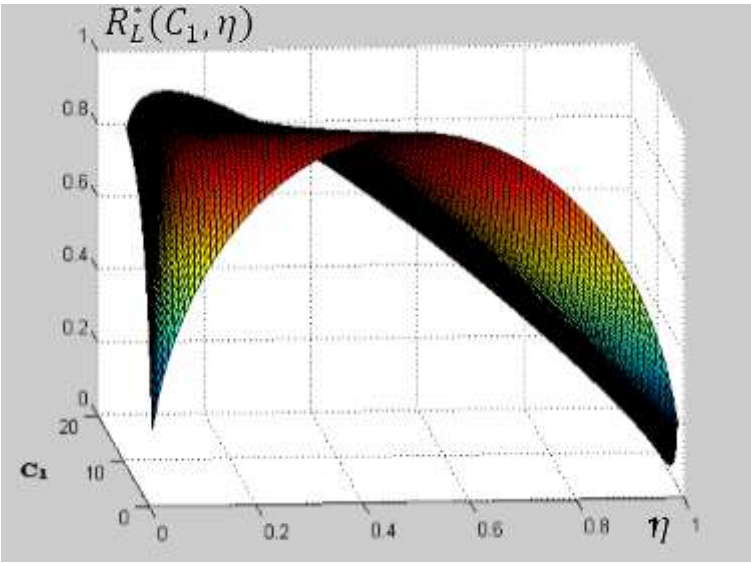


Figure 1: The concave shape of minimum conditional risk function $R^*_L(C_1,\eta)$.

Next the derivation of ICS boosting extensions by the gradient descent to the ICS exponential loss function is provided for AdaBoost, RealBoost and GentleBoost.

### 2.3. Deriving the formulas for the Instance-based Cost Sensitive boosting algorithm

AdaBoost, RealBoost and GentleBoost are the most popular classification algorithms in boosting framework which tries to minimize exponential loss function. In this part, the derivation of this classifier is demonstrated by using ICS exponential loss function.

#### 2.3.1. ICSAdaBoost:

Gradient decent is applied to minimize the ICS-exponential the loss function of Eq. (18) based on the training samples $\{(x_i, y_i)\}_{i=1}^{n}$ and define two sets

$$I_+ = \{i | y_i = 1\}, \quad I_- = \{i | y_i = -1\}. \tag{22}$$

$\alpha_m$ and $g_m$ are selected at iteration $m$ by gradient decent.

$$(\alpha_m, g_m) = \arg\min_{\alpha, f} \sum_{i \in I_+} w_i^{(m)} \exp(-C_{X,Y}(\mathrm{x}, 1)\alpha g(x_i)) + \sum_{i \in I_-} w_i^{(m)} \exp(C_{X,Y}(\mathrm{x}, -1)\alpha g(x_i)). \tag{23}$$

with

$$w_i^{(m+1)} = \begin{cases} w_i^{(m)} \exp(-C_{X,Y}(\mathrm{x}, 1)\alpha_m g_m(x_i)), & i \in \mathrm{I}_+ \\ w_i^{(m)} \exp(C_{X,Y}(\mathrm{x}, -1)\alpha_m g_m(x_i)), & i \in \mathrm{I}_- \end{cases} \tag{24}$$

Solution of Eq. (25) is the optimum $\alpha$ step.

$$2\bar{C}_{X,Y}(x, 1).b.\cosh\left(\bar{C}_{X,Y}(x, 1)\alpha\right) + 2\bar{C}_{X,Y}(x, -1).d.\cosh\left(\bar{C}_{X,Y}(x, -1)\alpha\right) = \bar{C}_{X,Y}(x, 1).\mathcal{T}_+.e^{-\bar{C}_{X,Y}(x,1)\alpha} + \bar{C}_{X,Y}(x, -1).\mathcal{T}_-.e^{-\bar{C}_{X,Y}(x,-1)\alpha} \tag{25}$$

where $\bar{C}_{X,Y}(x, y)$ is the average of $C_{X,Y}(x, y)$ and

$$\mathcal{T}_+ = \sum_{i \in I_+} w_i^{(m)}, \quad \mathcal{T}_- = \sum_{i \in I_-} w_i^{(m)}. \tag{26}$$

$$b = \sum_{i \in I_+} w_i^{(m)} [1 - I(y_i = g(x_i)], \quad d = \sum_{i \in I_-} w_i^{(m)} [1 - I(y_i = g(x_i)], \tag{27}$$

and the direction of gradient decent is given by Eq. (28).

$$g_m = \arg\min_g \left[\left(\exp\left(C_{X,Y}(\mathrm{x}, 1)\alpha g\right) - \exp\left(-C_{X,Y}(\mathrm{x}, 1)\alpha g\right)\right).b + \exp\left(-C_{X,Y}(\mathrm{x}, 1)\alpha g\right)\mathcal{T}_+ + \left(\exp\left(C_{X,Y}(\mathrm{x}, -1)\alpha g\right) - \exp\left(-C_{X,Y}(\mathrm{x}, -1)\alpha g\right)\right).d + \exp\left(-C_{X,Y}(\mathrm{x}, -1)\alpha g\right)\mathcal{T}_-\right]. \tag{28}$$

**Proof:** see Appendix C

Just like AdaBoost in ICSAdaBoost algorithm, there is a set of binary weak learners $\{g_k\}_{k=1}^{K}$ which are the possible directions of gradient decent. At each iteration cycles throw them and try to solve Eq. (25) by every scalar search procedure such as bisection search. Considering the same cost for the instances ($C_{X,Y}(\mathrm{x}, \mathrm{y}) = 1$), ICSAdaBoost reduces to AdaBoost. ICSAdaBoost is presented in following algorithm.

**Algorithm 1. ICSAdaBoost**

Input: Training set $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $y_i \epsilon \{1, -1\}$, Cost function $C_{X,Y}(\mathrm{x}, \mathrm{y})$, A set of binary weak learners $\{g_k\}_{k=1}^{K}$and M, the number of weak learners in final decision rule.

1. Start with weights $w_i^{(1)} = \frac{1}{n}$, $i = 1, 2, \ldots, n$.
2. Repeat for m=1, 2, …, M:
    2.1.Repeat for k=1, 2,…, K:
        2.1.1. Considering $g(x) = g_k(x)$ and compute (26), (27) and find the proper $\alpha$ according to Eq. (25).
        2.1.2. Compute (28) to attain the loss of weak learner $(g_k(x), \alpha_k)$.
    2.2. Select weak learner $(g_m(x), \alpha_m)$ according to the smallest loss.

2.3.Update weights according to (24).

Output: decision rule is (12).

### 2.3.2. ICSRealBoost:

Weak learners $G_m(x)$ are real functions and there is a dictionary of features $\{\phi_k(x)\}_{k=1}^{K}$. Gradient decent is applied to minimize the ICS-exponential the loss function of Eq. (18) based on the training samples $\{(x_i, y_i)\}_{i=1}^{n}$. The largest descent step at each iteration is

$$G_m^{real}(x) = G_{\varphi_{k^*}}(x), \tag{29}$$

where the optimum feature is selected according to (30)

$$k^* = \arg\min_k \sum_{i\,\epsilon I_+} w_i^{(m)} \exp(-c_{X,Y}(x_i, 1) G_{\varphi_k}(x_i)) + \sum_{i\,\epsilon I_-} w_i^{(m)} \exp(c_{X,Y}(x_i, -1) G_{\varphi_k}(x_i)). \tag{30}$$

And the weights are updated by

$$w_i^{(m+1)} = \begin{cases} w_i^{(m)} \exp(-c_{X,Y}(x_i, 1) G_m^{real}(x_i)), & i \in \mathrm{I}_+ \\ w_i^{(m)} \exp(c_{X,Y}(x_i, -1) G_m^{real}(x_i)), & i \in \mathrm{I}_- \end{cases}, \tag{31}$$

and

$$G_\varphi(x) = \frac{1}{c_{X,Y}(x,1) + c_{X,Y}(x,-1)} \log \frac{P_{Y|X}^{(w)}(y = 1|\varphi(x)) c_{X,Y}(x,1)}{P_{Y|X}^{(w)}(y = -1|\varphi(x)) c_{X,Y}(x,-1)}, \tag{32}$$

$P_{Y|X}^{(w)}(y|\varphi(x))$ , y ∈ {-1, 1} are the probability estimation of samples under feature transformation ϕ(x) after weighting update of $w_i^{(m)}$.

**Proof .** see Appendix D.

The posterior probability $P_{Y|X}^{(w)}(y|\varphi(x))$, y ∈ {-1, 1} of each class can be estimated by different methods such as the ones that are introduced in [23]. Weighted histogram is applied because it does not impose any consideration on dataset. In order to avoid having empty bin histogram, regularization is done. It is noticeable that ICSReal algorithm reduces to RealBoost if $c_{X,Y}(x, y) = 1$ for all samples. ICSRealBoost is presented in the following algorithm.

**Algorithm 2. ICSRealBoost**

Input: Training set $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $y_i \epsilon \{1, -1\}$, cost function $C_{X,Y}(\mathrm{x}, \mathrm{y})$ and *M*, the number of weak learners in final decision rule.

1. Start with weights $w_i^{(1)} = \frac{1}{n}$, $i = 1, 2, \ldots, n$.
2. Repeat for m=1, 2, …, M:
   2.1. Repeat for k=1, 2,…, K:

2.1.1. Compute the $G_{\varphi_k}(x)$ using (32).

2.2. Select the weak learner $G_m^{real}(x)$ according to (29) which has the smallest loss (30).

2.3. Update weights according to (31).

Output: decision rule is $h(x) = sign(\sum_{m=1}^{M} G_m^{real}(\mathrm{x}))$

Instances are classified based on (6), and to determine the costs, we need to calculate $G_{\varphi}(x)$ function, which are not known in the test phase; therefore, two regression models are trained to predict cost functions. Since every standard regression model [50, 51] can be used, we applied regression tree which split criterion is Gini's diversity index [52].

### 2.3.3. ICSGentleBoost

Weak learners $G_m(x)$ are real functions and there is a dictionary of features $\{\phi_k(x)\}_{k=1}^{K}$. Newton-Raphson is applied to minimize the ICS-exponential the loss function of Eq. (18) based on the training samples $\{(x_i, y_i)\}_{i=1}^{n}$. The Newton-Raphson step at iteration m is

$$G_m^{Gentle}(x) = G_{\phi_{k^*}}(x), \tag{33}$$

where $G_{\phi}(x) = a_{\phi}\phi(x) + b_{\phi}$ is the result of weighted regression.

$$(a_{\phi}, b_{\phi}) = \arg\min_{a_{\phi}, b_{\phi}} \sum_i w_i^{(m)} (y'_i - a_{\phi}\phi(x_i) - b_{\phi})^2 \tag{34}$$

with

$$y'_i = \frac{y_i}{C_{X,Y}(x_i, y_i)} \tag{35}$$

$$w_i^{(m)} = C_{X,Y}(x_i, y_i)^2 e^{-y_i C_{X,Y}(x_i, y_i)\hat{f}^{(m)}(x_i)} \tag{36}$$

where $\hat{f}^{(m)}(x)$ is the ICSGentleBoost predictor untill iteration *m*. The optimum feature is determined by

$$k^* = \arg\min_k \sum_i w_i^{(m)} (y'_i - a_{\phi_k}\phi_k(x_i) - b_{\phi_k})^2 \tag{37}$$

**Proof:** see Appendix E.

A summery of ICSGentleBoost is given in algorithm 3. The algorithm is fully compatible with GentleBoost when $C_{X,Y}(x, y) = 1$.

**Algorithm 3. ICSGentleBoost**

Input: Training set $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $y_i \epsilon \{1, -1\}$, cost function $C_{X,Y}(\mathrm{x}, \mathrm{y})$ and *M* is the number of weak learners in final decision rule.

1. Start with $\hat{f}^{(1)}(x_i) = 0., y'_i = \frac{y_i}{C_{X,Y}(x_i, y_i)}, \ i = 1, 2, \ldots, n.$
2. Repeat for m=1, 2, …, M:
   2.1. Update weights according to (36).

2.2.Repeat for k=1, 2,…, K:

2.2.1. Compute the solution to the least square problem of (34),

$$a_{\phi_k} = \frac{\langle 1\rangle_w.\langle \phi_k(\mathrm{x}_i) y'_i\rangle_w - \langle \phi_k(\mathrm{x}_i)\rangle_w.\langle y'_i\rangle_w}{\langle 1\rangle_w.\langle \phi_k^2(\mathrm{x}_i)\rangle_w - \langle \phi_k(\mathrm{x}_i)\rangle_w^2} \quad (38)$$

$$b_{\phi_k} = \frac{\langle \phi_k^2(\mathrm{x}_i)\rangle_w.\langle y'_i\rangle_w - \langle \phi_k(\mathrm{x}_i)\rangle_w.\langle \phi_k(\mathrm{x}_i) y'_i\rangle_w}{\langle 1\rangle_w.\langle \phi_k^2(\mathrm{x}_i)\rangle_w - \langle \phi_k(\mathrm{x}_i)\rangle_w^2} \quad (39)$$

where we have defined

$$\langle q(\mathrm{x}_i)\rangle_w \stackrel{\text{def}}{=} \sum_i w_i^{(m)} q(x_i) \quad (40)$$

2.3. Select the optimum direction according to (37) and set the weak learner $G_m^{Gentle}(x)$ according to (33).

2.4. Set $\hat{f}^{(m+1)} = \hat{f}^{(m)} + G_m^{Gentle}(\mathrm{x})$.

Output: The decision rule is $h(x) = sign(\sum_{m=1}^{M} G_m^{Gentle}(\mathrm{x})$.

## Generative Cost Sensitive Function

The proposed cost function in this study benefits from both experimental and mathematical based approaches. Cost of each sample is calculated by summation of two terms where the first one is an experimental cost value and the second one is specified by its importance in the class distribution density. Although we discussed about disadvantages of considering the same cost value for all the samples in Section 1, this constant value can be interpreted as an average cost ratio between the classes. There is no doubt that samples of the positive class have a higher cost than those belonging to the insensitive class, incidentally, the cost of all samples should not be equal; consequently, the costs of samples are fluctuating above this average value. To estimate a meaningful value representing the cost of each instance, first the probability density function (PDF) of its corresponding class is determined using Gaussian Mixture Model (GMM) [1] which is a flexible model capable of being fitted to every arbitrary shape. Then, its probability is easily determined that constructs the second term. According to this explanation, the cost of each sample is determined as follows:

$$C_{X,Y}(x, y) = C_y + P_{X|Y}(x|y), \;\; y \in \{1, -1\} \quad (41)$$

Where $C_y$ is the average costs of instances belonging to the positive and negative classes $(C_1, C_{-1})$, respectively and $P_{X|Y}(x|y)$ is the probability class density function (PDF) that is estimated for each class separately. This idea seems logical because marginal samples carry lower cost in comparison with the samples located in the area with higher probability value. In other words, the reliable samples are located around the mean distribution while noisy and outlier samples are placed in borders.

## Dataset

Twelve dataset from cost sensitive scopes such as life and game, in UCI repository database are selected; however, costs of instances are not reported in datasets, hence, by employing the proposed cost generation schemes introduced in Eq. (20), the cost of each instance is determined for the selected datasets. Instances with missing value are excluded from the datasets because of using the decision stump as weak learner.

As a preprocessing stage, some modifications are applied to the employed datasets to be more proper for classification. Id number feature from Breast Cancer Wisconsin is omitted causing it does not give any discriminative information. SPECTF consists of two parts: train and test sets, the train and test samples are blended to better evaluate the classifier performance in more testing folds. The multiclass dataset of Cardio-tocography is converted to a two class problems 1) by attaining the suspected and pathologic classes (removing the instances of normal group) named as Cardiotocography_S-P and 2) by gathering the suspected and pathologic classes into one class verses normal class named Cardiotocography_N-SP. Table 2, describes the selected datasets in terms of number of samples, number of attributes, and the population ratio of high-cost/low-cost classes.

As it shown in Table 2, the selected datasets cover a wide range of applications. Hepatits, WPBC and SpectF Heart have low number of samples and large number of features; Bupa Liver, Breast Cancer Wisconsin, Pima Indians and Tic-Tac-Toe have proper number of sample and feature for classification algorithms and also large datasets such as Bank, Spam and Musk1 are employed to assess the proposed methods in different situations. From the population ratio of positive samples to negative samples selected datasets are spread into (0.1, 0.8).

## Experimental and result

To evaluate the performance of classifiers, the proposed methods and the state of art cost sensitive boosting classifiers (CSB0, CSB1, CSB2, AdaC1, AdaC2, AdaC3 and AdaCost) are implemented and applied to several imbalance datasets from UCI database. Decision stump is used as the weak learner in cost sensitive extension of AdaBoost algorithm. As far as the proposed cost function for each specific class contains two terms (a constant term and probability function), to find a proper value for the constant term, several discrete values are assigned to $C_1$ (the cost of positive class) such that $C_1 = \{2, 3, \dots, 10\}$ and $C_{-1}$ is considered +1. For each value of $C_1$ ten times ten folds cross validation is executed and accuracy, CPE (Cost Per Example) and F-measure are calculated for each dataset. Besides paired T-test, with t = 0.001 which is used in biomedical applications, is applied to the results and in the case of supremacy the value is boldface.

Table 3 shows the average accuracy for all datasets and classifiers. To simplify the comparison, the last column of Table 3, shows the number of datasets which each classifier has the highest accuracy, named as the number of wins. As we see, in seven datasets the proposed ICS algorithms outperform the other compared methods; however, in the other six datasets AdaCost and AdaC1 appear better than the proposed algorithms. The supremacy of AdaCost and AdaC1 in terms of accuracy is the result of trying to maximize the accuracy without giving proper attention to the examples cost which can be obvious in terms of CPE and F-measure.

The average CPE for all datasets and classifiers are reported in Table 4. In the most cases, our proposed algorithms outperform the others except for SPECTF-Heart and Hepatitis, CSB0 and CSB2 do better than our methods. Looking at Tables 3 and 4 more carefully, we can find the answer why in these datasets the proposed ICS classifiers could not outperform the others. For Hepatitis, CPE different between CSB2 and our methods is small about (1.13, 3.56) while in the case of accuracy our classifier really do better about (10.16, 20.65). The same story is happened between CSB0 and the proposed ICS classifiers for SPECTF-Heart dataset. It shows that the proposed methods make a better balance between cost and accuracy, in other words, they predict the cost sensitive boundary more accurately. Table 4 demonstrates the supremacy of the proposed methods in the case of using F-measure. Regarding F-measure which is the best/fair measurement for classification supremacy of cost sensitive/ imbalance dataset, ICSGentleboost get the first rank, while ICSRealboost, ICSAdaBoost maintain the subsequent positions, respectively.

## Conclusion

In this paper a Bayesian consistent instance based cost sensitive exponential loss function is proposed. Based on the proposed loss function, the derivation of instance cost sensitive extensions of AdaBoost, RealBoost and GentleBoost are developed. In order to evaluate the performance of the proposed methods on datasets, we determine the example cost based on an empirical class cost and a density based approach. Experimental results on thirteen UCI datasets is presented and shown that our methods among the various prior cost sensitive methods (CSB0, CSB1, CSB2, AdaCost, AdaC1, AdaC2, AdaC3) perform well. The future work will be on designing other example dependent cost sensitive loss function algorithm with real application and example costs.

Table 2: Statistics of Eight UCI Datasets.

| # | Dataset | #Sample | #feature | $\frac{\#positive}{\#negative}$ |
|---|---|---|---|---|
| 1 | Hepatitis | 100 | 19 | 13/67 |
| 2 | WPBC | 194 | 33 | 46/148 |
| 3 | SPECTF Heart | 267 | 44 | 55/212 |
| 4 | Bupa Liver | 345 | 6 | 145/200 |
| 5 | Cardiotocography-S-P | 471 | 21 | 176/295 |
| 6 | WDBC | 569 | 30 | 212/357 |
| 7 | Breast Cancer Wisconsin | 683 | 9 | 239/444 |

| | | | | |
|---|---|---|---|---|
| 8 | Pima Indians | 768 | 7 | 268/500 |
| 9 | Tic-Tac-Toe | 958 | 9 | 332/626 |
| 10 | Cardiotocography-N,SP | 2126 | 21 | 471/1655 |
| 11 | Bank | 4521 | 16 | 521/4000 |
| 12 | Spam | 4601 | 57 | 1813/2788 |
| 13 | Musk1 | 7074 | 166 | 1224/5850 |

Table 2. Test Accuracy

| Test ACC | Hepatitis | WPBC | SPECTF Heart | Bupa Liver | Cardiotocography-S-P | WDBC | Breast Cancer Wisconsin |
|---|---|---|---|---|---|---|---|
| **CSB0** | 77.49 | 63.99 | 65.00 | 44.18 | 70.46 | 90.24 | 89.35 |
| **CSB1** | 61.96 | 29.28 | 34.35 | 40.50 | 39.914 | 80.89 | 42.94 |
| **CSB2** | 68.14 | 29.1 | 34.74 | 41.24 | 40.66 | 85.27 | 45.12 |
| **AdaCost** | 86.19 | **77.74** | **81.62** | **62.376** | 82.37 | 93.62 | 92.08 |
| **Adac1** | 83.74 | 65.11 | 67.66 | 43.34 | 86.47 | 91.93 | 94.48 |
| **AdaC2** | 33.28 | 29.11 | 45.80 | 42.33 | 38.00 | 64.49 | 35.43 |
| **AdaC3** | 42.65 | 38.37 | 47.07 | 41.46 | 38.61 | 65.22 | 35.30 |
| **ICSReal** | 78.03 | 66.27 | 66.87 | 44.44 | **93.27** | 94.72 | 96.66 |
| **ICSAda** | 87.85 | 68.38 | 78.97 | 52.02 | 92.78 | **95.88** | 96.47 |
| **ICSGentle** | **88.79** | 71.59 | 77.26 | 48.12 | 81.22 | 95.51 | **96.98** |

| Test ACC | Pima Indians | Tic-Tac-Toe | Cardiotocography-N,SP | Bank | Spam | Musk1 | # Win in 13 dataset |
|---|---|---|---|---|---|---|---|
| **CSB0** | 58.21 | 44.28 | 84.54 | 76.04 | 68.97 | 75.01 | 0 |
| **CSB1** | 34.66 | 33.19 | 41.12 | 11.74 | 39.97 | 18.69 | 0 |
| **CSB2** | 34.33 | 34.74 | 47.82 | 12.79 | 28.75 | 34.33 | 0 |
| **AdaCost** | **68.09** | **66.63** | 85.28 | 88.51 | 83.77 | 84.42 | 5 |
| **Adac1** | 57.00 | 48.29 | 87.75 | **88.90** | 82.36 | 76.58 | 1 |
| **AdaC2** | 38.52 | 34.74 | 50.89 | 17.25 | 39.89 | 19.88 | 0 |
| **AdaC3** | 34.56 | 34.74 | 50.95 | 59.52 | 42.42 | 21.36 | 0 |
| **ICSReal** | 57.76 | 65.72 | 84.44 | 66.73 | 74.31 | **86.50** | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **ICSAda** | 61.72 | 57.60 | 87.55 | 51.13 | **86.89** | 75.32 | 2 |
| **ICSGentle** | 64.86 | 66.39 | **88.19** | 83.84 | 81.90 | 82.62 | 3 |

Table 3 Test CPE

| **Test CPE** | **Hepatitis** | **WPBC** | **SPECTF Heart** | **Bupa Liver** | **Cardiotocography-S-P** | **WDBC** | **Breast Cancer Wisconsin** |
|---|---|---|---|---|---|---|---|
| **CSB0** | 75.95 | 66.85 | **76.14** | 77.97 | 85.36 | 92.11 | 93.82 |
| **CSB1** | 75.12 | 50.64 | 64.36 | 74.46 | 76.47 | 89.75 | 77.37 |
| **CSB2** | **77.45** | 52.82 | 72.94 | 77.80 | 76.81 | 91.92 | 78.76 |
| **AdaCost** | 57.39 | 52.95 | 47.46 | 29.23 | 64.30 | 88.66 | 83.81 |
| **Adac1** | 73.24 | 66.78 | 69.47 | 76.43 | 77.99 | 90.30 | 92.80 |
| **AdaC2** | 56.34 | 52.72 | 68.05 | 77.78 | 75.03 | 80.12 | 72.74 |
| **AdaC3** | 56.56 | 57.93 | 68.96 | 77.28 | 75.34 | 80.65 | 72.72 |
| **ICSReal** | 73.89 | 61.24 | 71.84 | 78.33 | **93.99** | **94.98** | 96.18 |
| **ICSAda** | 75.78 | 64.00 | 72.61 | **78.50** | 93.24 | 94.51 | 95.78 |
| **ICSGentle** | 76.32 | **69.41** | 73.25 | 78.23 | 87.57 | 93.39 | **96.59** |
| | | | | | | | |
| **Test CPE** | **Pima Indians** | **Tic-Tac-Toe** | **Cardiotocography-N,SP** | **Bank** | **Spam** | **Musk1** | **# Win in 13 dataset** |
| **CSB0** | 79.56 | 75.25 | 85.11 | 75.57 | 88.83 | 80.62 | 1 |
| **CSB1** | 72.70 | 63.25 | 60.09 | 40.84 | 79.46 | 53.87 | 0 |
| **CSB2** | 72.60 | 72.94 | 66.08 | 42.71 | 66.17 | 72.60 | 1 |
| **AdaCost** | 33.74 | 30.00 | 69.03 | 58.83 | 68.16 | 52.21 | 0 |
| **Adac1** | 76.19 | 73.52 | 80.35 | 72.57 | 82.73 | 71.11 | 0 |
| **AdaC2** | 73.36 | 72.94 | 67.10 | 45.39 | 79.53 | 54.21 | 0 |
| **AdaC3** | 72.30 | 72.94 | 67.15 | 65.31 | 80.66 | 55.03 | 0 |
| **ICSReal** | 80.86 | **86.33** | 88.02 | 75.99 | 88.24 | **90.88** | 4 |
| **ICSAda** | 79.89 | 72.70 | **90.24** | 67.40 | **93.69** | 84.68 | 3 |
| **ICSGentle** | **81.51** | 83.78 | 89.60 | **80.03** | 92.64 | 85.16 | 4 |

Table 4. F-measure

| F-measure | Hepatitis | WPBC | SPECTF Heart | Bupa Liver | Cardiotocography-S-P | WDBC | Breast Cancer Wisconsin |
|---|---|---|---|---|---|---|---|
| **CSB0** | 0.4622 | 0.4377 | 0.4822 | 0.5888 | 0.7100 | 0.8800 | 0.8633 |
| **CSB1** | 0.4322 | 0.3500 | 0.3711 | 0.5611 | 0.5522 | 0.8000 | 0.5566 |
| **CSB2** | 0.4644 | 0.3744 | 0.3944 | 0.5800 | 0.5522 | 0.8377 | 0.5655 |
| **AdaCost** | 0.1311 | 0.0677 | 0.0988 | 0.1566 | 0.6722 | 0.9066 | 0.8688 |
| **Adac1** | 0.4177 | 0.4033 | 0.4333 | 0.5733 | 0.7888 | 0.8888 | 0.9155 |
| **AdaC2** | 0.2733 | 0.3822 | 0.4055 | 0.5833 | 0.5455 | 0.6766 | 0.5133 |
| **AdaC3** | 0.2611 | 0.4155 | 0.4111 | 0.5811 | 0.5466 | 0.6888 | 0.5155 |
| **ICSReal** | 0.4933 | 0.3755 | 0.4688 | 0.5877 | **0.8466** | 0.9333 | 0.9511 |
| **ICSAda** | 0.2800 | **0.4911** | 0.3977 | 0.5177 | 0.6788 | 0.8888 | 0.6066 |
| **ICSGentle** | **0.4988** | 0.4900 | **0.5311** | **0.600** | 0.7844 | **0.9455** | **0.9588** |
| | | | | | | | |
| **F-measure** | **Pima Indians** | **Tic-Tac-Toe** | **Cardiotocography-N,SP** | **Bank** | **Spam** | **Musk1** | **# Win in 13 dataset** |
| **CSB0** | 0.6000 | 0.5366 | 0.6922 | 0.3888 | 0.7177 | 0.5300 | 0 |
| **CSB1** | 0.5111 | 0.4433 | 0.4200 | 0.2011 | 0.5700 | 0.2955 | 0 |
| **CSB2** | 0.5100 | 0.5200 | 0.4800 | 0.2111 | 0.5733 | 0.2955 | 0 |
| **AdaCost** | 0.1611 | 0.0711 | 0.4944 | 0.0277 | 0.7422 | 0.1788 | 0 |
| **Adac1** | 0.5800 | 0.5355 | 0.6755 | 0.4322 | 0.7855 | 0.4644 | 0 |
| **AdaC2** | 0.5255 | 0.5200 | 0.4866 | 0.2200 | 0.5711 | 0.3000 | 0 |
| **AdaC3** | 0.5100 | 0.5200 | 0.4877 | 0.3433 | 0.5788 | 0.3055 | 0 |
| **ICSReal** | 0.6100 | **0.6688** | 0.7144 | 0.3144 | 0.7455 | **0.7111** | 3 |
| **ICSAda** | 0.5055 | 0.5466 | 0.5933 | 0.2933 | 0.5244 | 0.4266 | 1 |
| **ICSGentle** | **0.6377** | 0.6611 | **0.7677** | **0.4900** | **0.8111** | 0.6344 | 9 |

## Appendix A

Considering Eq. (5) as the loss function, risk of classifier $h(x)$ deciding an instance positive or negative will be

$$R(h(\mathrm{x}) = 1|\mathrm{x}) = C_{X,Y}(\mathrm{x}, -1)\mathrm{P}_{\mathrm{Y}|\mathrm{X}}(-1|\mathrm{x})$$

$$R(h(\mathrm{x}) = -1|\mathrm{x}) = C_{X,Y}(\mathrm{x}, 1)\mathrm{P}_{\mathrm{Y}|\mathrm{X}}(1|\mathrm{x})$$

If $R(h(\mathrm{x})=1|\mathrm{x}) < R(h(\mathrm{x})=-1|\mathrm{x})$ then based on BDR the instance classify as positive

$C_{X,Y}(\mathrm{x},-1)\mathrm{P}_{\mathrm{Y|X}}(-1|\mathrm{x}) < C_{X,Y}(\mathrm{x},1)\mathrm{P}_{\mathrm{Y|X}}(1|\mathrm{x})$

Divide both side by $C_{X,Y}(\mathrm{x},-1)\mathrm{P}_{\mathrm{Y|X}}(-1|\mathrm{x})$

$$1 < \frac{C_{X,Y}(\mathrm{x},1)P_{Y|X}(\,1 \mid \mathrm{x})}{C_{X,Y}(\mathrm{x},-1)P_{Y|X}(-1 \mid \mathrm{x})}$$

$$1 < \frac{P_{Y|X}^{(c)}(1|x)}{P_{Y|X}^{(c)}(-1|x)} \qquad (*)$$

Where

$$P_{Y|X}^{(c)}(y'|X) = \frac{C_{X,Y}(\mathrm{x},y')P_{Y|X}(y' \mid \mathrm{x})}{\sum_{y\in\{-1,1\}} C_{X,Y}(\mathrm{x},\mathrm{y})P_{Y|X}(\mathrm{y} \mid \mathrm{x})} \qquad (**)$$

Taking logarithm from both side of equation (*) and then applying the sign function, attaining the optimum BDR.

$$h^*(x) = sign\left[\log\frac{P_{Y|X}^{(c)}(1|x)}{P_{Y|X}^{(c)}(-1|x)}\right].$$

Appendix B

The instance cost sensitive exponential loss function () is minimized by minimizing the expectation of loss function conditioned on x.

$$\begin{aligned} \mathrm{L}(\mathrm{x}) &= E_{X,Y}[I(y=1)e^{-yC_{X,Y}(\mathrm{x},1)f(\mathrm{x})} + I(y=-1)e^{-yC_{X,Y}(\mathrm{x},-1)f(\mathrm{x})}] \\ &= P_{Y|X}(1\,|x)e^{-C_{X,Y}(\mathrm{x},1)f(x)} + P_{Y|X}(-1\,|x)e^{C_{X,Y}(\mathrm{x},-1)f(x)}\,. \end{aligned}$$

Setting derivative to zero

$$\frac{dL(x)}{df(x)} = -C_{X,Y}(\mathrm{x},1)P_{Y|X}(1\,|x)e^{-C_{X,Y}(\mathrm{x},1)f(x)} + C_{X,Y}(\mathrm{x},-1)P_{Y|X}(-1\,|x)e^{C_{X,Y}(\mathrm{x},-1)f(x)} = 0,$$

It follows that

$$\frac{C_{X,Y}(\mathrm{x},1)P_{Y|X}(1\,|x)}{C_{X,Y}(\mathrm{x},-1)P_{Y|X}(-1\,|x)} = e^{(C_{X,Y}(\mathrm{x},1)+C_{X,Y}(\mathrm{x},-1))f(x)}$$

and

$$f(x) = \frac{1}{C_{X,Y}(\mathrm{x},1)+C_{X,Y}(\mathrm{x},-1)}\log\frac{P_{Y|X}(\,1|x)C_{X,Y}(\mathrm{x},1)}{P_{Y|X}(\,-1|x)C_{X,Y}(\mathrm{x},-1)}.$$

Dividing the numerator and denominator of the logarithm by $\sum_{y\in\{-1,1\}} C_{X,Y}(\mathrm{x},\mathrm{y})P_{Y|X}(\mathrm{y} \mid \mathrm{x})$ and using (),

$$f(x) = \frac{1}{C_{X,Y}(\mathrm{x},1) + C_{X,Y}(\mathrm{x},-1)} \log \frac{P_{Y|X}(\,1|x) C_{X,Y}(\mathrm{x},1)}{P_{Y|X}(\,-1|x) C_{X,Y}(\mathrm{x},-1)}$$

Second derivative is nonnegative, so the loss is minimized by $f(x)$.

Appendix C

$$J(f) = E_{X,Y}[I(y=1)e^{-C_{X,Y}(\mathrm{x},1)f(x)} + I(y=-1)e^{C_{X,Y}((x,-1)f(x)}]$$

By adding $\alpha g(x)$ to $f(x)$, we have

$$J(f+\alpha g) = E_{X,Y}[I(y=1)w(x,1)e^{-C_{X,Y}(\mathrm{x},1)\alpha g(x)} + I(y=-1)w(x,-1)e^{C_{X,Y}(\mathrm{x},-1)\alpha g(x)}]$$

Where $w(x,y) = e^{-yC_{X,Y}(\mathrm{x},\mathrm{y})f(x)}$.

The expectation will be minimized when it is minimized by all x.

$$\left(\alpha_m g_m(\mathrm{x})\right) = \arg\min_{\alpha, g(\mathrm{x})} E_{Y|X}\left[\begin{matrix} I(y=1)w(\mathrm{x},1)e^{-C_{X,Y}(\mathrm{x},1)\alpha g(\mathrm{x})} + \\ I(y=-1)w(\mathrm{x},-1)e^{C_{X,Y}(\mathrm{x},-1)\alpha g(x)} \end{matrix}\middle|\mathrm{x}\right].$$

$$E_{Y|X}[\,I(y=1)w(\mathrm{x},1)e^{-C_{X,Y}(\mathrm{x},1)\alpha g(\mathrm{x})} + I(y=-1)w(\mathrm{x},-1)e^{C_{X,Y}(\mathrm{x},-1)\alpha g(\mathrm{x})}|x]$$

$$\begin{aligned} E_{Y|X}[I(y=1)I(g(\mathrm{x})=1)w(\mathrm{x},1)e^{-C_{X,Y}(\mathrm{x},1)\alpha} &+ I(y=1)I(g(\mathrm{x})=-1)w(\mathrm{x},1)e^{C_{X,Y}(\mathrm{x},1)\alpha} \\ &+ I(y=-1)I(g(\mathrm{x})=1)w(\mathrm{x},-1)e^{C_{X,Y}(\mathrm{x},-1)\alpha} \\ &+ I(y=-1)I(g(\mathrm{x})=-1)w(\mathrm{x},-1)e^{-C_{X,Y}(\mathrm{x},-1)\alpha}|\mathrm{x}]. \end{aligned}$$

$$\begin{aligned} E_{Y|X}[I(y=1)I(g(\mathrm{x})=-1)w(\mathrm{x},1)(e^{C_{X,Y}(\mathrm{x},1)\alpha} - e^{-C_{X,Y}(\mathrm{x},1)\alpha}) &+ I(y=1)w(\mathrm{x},1)e^{-C_{X,Y}(\mathrm{x},1)\alpha} \\ &+ I(y=-1)I(g(\mathrm{x})=1)w(\mathrm{x},-1)(e^{C_{X,Y}(\mathrm{x},-1)\alpha} - e^{-C_{X,Y}(\mathrm{x},-1)\alpha}) \\ &+ I(y=-1)w(\mathrm{x},-1)e^{-C_{X,Y}(\mathrm{x},-1)\alpha}|\mathrm{x}] \end{aligned}$$

$$\begin{aligned} = P_{Y|X}(1|\mathrm{x})w(\mathrm{x},1)I(g(\mathrm{x})=-1)(e^{C_{X,Y}(\mathrm{x},1)\alpha} - e^{-C_{X,Y}(\mathrm{x},1)\alpha}) &+ P_{Y|X}(1|\mathrm{x})w(\mathrm{x},1)e^{-C_{X,Y}(\mathrm{x},1)\alpha} \\ &+ P_{Y|X}(-1|\mathrm{x})w(\mathrm{x},-1)I(g(\mathrm{x})=1)(e^{C_{X,Y}(\mathrm{x},-1)\alpha} - e^{-C_{X,Y}(\mathrm{x},-1)\alpha}) \\ &+ P_{Y|X}(-1|\mathrm{x})w(\mathrm{x},-1)e^{-C_{X,Y}(\mathrm{x},-1)\alpha}, \end{aligned}$$

We can write that

$$\begin{aligned} \left(\alpha_m g_m(\mathrm{x})\right) = \arg\min_{\alpha, g(x)} \{ &P_{Y|X}^{(w)}(1|\mathrm{x})I(g(\mathrm{x})=-1)(e^{C_{X,Y}(\mathrm{x},1)\alpha} - e^{-C_{X,Y}(\mathrm{x},1)\alpha}) \\ &+ P_{Y|X}^{(w)}(1|\mathrm{x})e^{-C_{X,Y}(\mathrm{x},1)\alpha} + P_{Y|X}^{(w)}(-1|\mathrm{x})I(g(\mathrm{x})=1)(e^{C_{X,Y}(\mathrm{x},-1)\alpha} - e^{-C_{X,Y}(\mathrm{x},-1)\alpha}) \\ &+ P_{Y|X}^{(w)}(-1|\mathrm{x})e^{-C_{X,Y}(\mathrm{x},-1)\alpha}\} \end{aligned}$$

Where $P_{Y|X}^{(w)}(y|\mathrm{x}) = \frac{P_{Y|X}(y|\mathrm{x})w(\mathrm{x},y)}{\sum_{y\epsilon\{-1,1\}} P_{Y|X}(y|\mathrm{x})w(\mathrm{x},y)}$ is posterior probability based on new samples weight and optimum weak learner is

$$\begin{aligned}\big(\alpha_m g_m(\mathrm{x})\big) = \arg\min_{\alpha, g(\mathrm{x})} \{ & P_{Y|X}^{(w)}(1|\mathrm{x}) I(g(\mathrm{x}) = -1)(e^{C_{X,Y}(\mathrm{x},1)\alpha} - e^{-C_{X,Y}(\mathrm{x},1)\alpha}) \\ & + P_{Y|X}^{(w)}(1|\mathrm{x}) e^{-C_{X,Y}(\mathrm{x},1)\alpha} + P_{Y|X}^{(w)}(-1|\mathrm{x}) I(g(\mathrm{x}) = 1)(e^{C_{X,Y}(\mathrm{x},-1)\alpha} - e^{-C_{X,Y}(\mathrm{x},-1)\alpha}) \\ & + P_{Y|X}^{(w)}(-1|\mathrm{x}) e^{-C_{X,Y}(\mathrm{x},-1)\alpha} \}\end{aligned}$$

Replacing expectation by average

$$\big(\alpha_m g_m(\mathrm{x})\big) = \arg\min_{\alpha, g(\mathrm{x})} (e^{\bar{C}_{X,Y}(x,1)\alpha} - e^{-\bar{C}_{X,Y}(x,1)\alpha}).\, b + \mathcal{T}_+.\, e^{-\bar{C}_{X,Y}(x,1)\alpha} + \big(e^{\bar{C}_{X,Y}(x,-1)\alpha} - e^{-\bar{C}_{X,Y}(x,-1)\alpha}\big).\, d + \mathcal{T}_-.\, e^{-\bar{C}_{X,Y}(x,-1)\alpha} | \mathrm{x}],$$

Using empirical estimate of $\mathcal{T}_-, \mathcal{T}_+, b, d$, and given $g(x)$ and setting the derivative with respect to $\alpha$ to zero,

$$\begin{aligned}\frac{\partial}{\partial \alpha} = & \bar{C}_{X,Y}(x,1)(e^{\bar{C}_{X,Y}(x,1)\alpha} - e^{-\bar{C}_{X,Y}(x,1)\alpha}).\, b - \mathcal{T}_+.\, \bar{C}_{X,Y}(x,1) e^{-\bar{C}_{X,Y}(x,1)\alpha} \\ & + \bar{C}_{X,Y}(x,-1)\big(e^{\bar{C}_{X,Y}(x,-1)\alpha} - e^{-\bar{C}_{X,Y}(x,-1)\alpha}\big).\, d - \bar{C}_{X,Y}(x,-1)\mathcal{T}_-.\, e^{-\bar{C}_{X,Y}(x,-1)\alpha} \\ & = 0,\end{aligned}$$

The solution of below equation is the optimum step size $\alpha$

$$\begin{aligned}2\bar{C}_{X,Y}(x,1).\, b.\, \cosh\big(\bar{C}_{X,Y}(x,1)\alpha\big) + 2\bar{C}_{X,Y}(x,-1).\, d.\, \cosh\big(\bar{C}_{X,Y}(x,-1)\alpha\big) \\ = \bar{C}_{X,Y}(x,1).\, \mathcal{T}_+.\, e^{-\bar{C}_{X,Y}(x,1)\alpha} + \bar{C}_{X,Y}(x,-1).\, \mathcal{T}_-.\, e^{-\bar{C}_{X,Y}(x,-1)\alpha}\end{aligned}$$

Appendix D

The loss function is

$$J[f] = E_{X,Y}[I(y = 1) e^{-C_{X,Y}(\mathrm{x},1) f(x)} + I(y = -1) e^{C_{X,Y}(\mathrm{x},-1) f(x)}]$$

By adding $G(x)$ to f(x)

$$J[f + G] = E_{X,Y}[I(y = 1) w(x,1) e^{-C_{X,Y}(\mathrm{x},1) G(x)} + I(y = -1) w(x,-1) e^{C_{X,Y}(\mathrm{x},-1) G(x)}]$$

Where $w(x,y) = e^{-C_{X,Y}(\mathrm{x},\mathrm{y}) f(x)}$

The expectation is minimized if it is minimized for all x, assume the weak learner only consider some feature of $\varphi(x)$. The optimum weak learner is the solution of

$$G_\varphi(x) = \arg\min_G E_{Y|X}\big[I(y = 1) w(x,1) e^{-C_{X,Y}(\mathrm{x},1) G(x)} + I(y = -1) w(x,-1) e^{C_{X,Y}(\mathrm{x},-1) G(x)} \big| x\big]$$

$$= \arg\min_G P_{Y|X}(1\,|\varphi(x)) w(x,1) e^{-C_{X,Y}(\mathrm{x},1) G(x)} + P_{Y|X}(-1\,|\varphi(x)) w(x,-1) e^{C_{X,Y}(\mathrm{x},-1) G(x)}$$

$$= \arg\min_G P_{Y|X}^{(w)}(1\,|\varphi(x)) e^{-C_{X,Y}(\mathrm{x},1) G(x)} + P_{Y|X}^{(w)}(-1\,|\varphi(x)) e^{C_{X,Y}(\mathrm{x},-1) G(x)}.$$

Where

$$P_{Y|X}^{(w)}(1\,|\varphi(x)) = \frac{P_{Y|X}(y\,|\varphi(x))w(x,y)}{\sum_{y\epsilon\{1,-1\}} P_{Y|X}(y\,|\varphi(x))w(x,y)}$$

Are the posterior probability of

$$G_{\varphi}(x) = \left\{\frac{1}{C_{X,Y}(\mathrm{x},1) + C_{X,Y}(\mathrm{x},-1)}\log\frac{P_{Y|X}^{(w)}(1|\phi(\mathrm{x}))C_{X,Y}(\mathrm{x},1)}{P_{Y|X}^{(w)}(-1|\phi(\mathrm{x}))C_{X,Y}(\mathrm{x},-1)}\right\}.$$

Divide numerator and dominator of logarithm term by $\sum_{y\in\{-1,1\}} C_{X,Y}(\mathrm{x},\mathrm{y})P_{Y|X}^{(w)}(\mathrm{y}\mid\phi(\mathrm{x}))$ and using (**) definition,

$$G_{\varphi}(x) = \left\{\frac{1}{C_{X,Y}(\mathrm{x},1) + C_{X,Y}(\mathrm{x},-1)}\log\frac{P_{Y|X}^{(w,C)}(1|\phi(\mathrm{x}))}{P_{Y|X}^{(w,C)}(-1|\phi(\mathrm{x}))}\right\}.$$

The optimum $\varphi$ minimized following equation

$$\varphi^* = \arg\min_{\varphi} J\left[F + G_{\varphi}\right]$$

$$= \arg\min_{\varphi} E_{Y|X}\left[I(y=1)w(x,1)e^{-C_{X,Y}(\mathrm{x},1)G_{\varphi}(x)} + I(y=-1)w(x,-1)e^{C_{X,Y}(\mathrm{x},-1)(x)}\middle|x\right]$$

$$= \arg\min_{\varphi}\left[\sum_{i\epsilon\, I_+} w(x_i,1)e^{-C_{X,Y}(x_i,1)G_{\varphi}(x_i)} + \sum_{i\epsilon\, I_-} w(x_i,-1)e^{C_{X,Y}(x_i,-1)G_{\varphi}(x_i)}\right].$$

Appendix E

Rewriting the instance cost sensitive exponential loss function,

$$l\left(\hat{f}^{(m)}(\mathrm{x})|\mathrm{x}\right) = E_{Y|X}\left[e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}|x\right]$$

We want ro minimized the loss function by adding $G(x)$,

$$l\left(\hat{f}^{(m)}(\mathrm{x}) + G(\mathrm{x})|\mathrm{x}\right) = E_{Y|X}\left[e^{-yC_{X,Y}(x,\mathrm{y})(\hat{f}^{(m)}(\mathrm{x})+G(\mathrm{x}))}|x\right]$$

ICSGentleBoost algorithm uses newton step to find optimum $G(x)$

First derivation

$$\frac{\partial J\left(\hat{f}^{(m)}(\mathrm{x})+G(\mathrm{x})\right)}{\partial G(\mathrm{x})}_{G(\mathrm{x})=0} = E_{Y|X}\left[-yC_{X,Y}(x,\mathrm{y})e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}|x\right]$$

Second derivation

$$\frac{\partial^2 J\left(\hat{f}^{(m)}(\mathrm{x})+G(\mathrm{x})\right)}{\partial G(\mathrm{x})^2}_{G(\mathrm{x})=0} = E_{Y|X}\left[C_{X,Y}(x,\mathrm{y})^2 e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}|x\right]$$

$$G(\mathrm{x}) = \frac{E_{Y|X}\left[-yC_{X,Y}(x,\mathrm{y})e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}|x\right]}{E_{Y|X}\left[C_{X,Y}(x,\mathrm{y})^2 e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}|x\right]}$$

By using the definition of weighed conditional expectation,

$$G(\mathrm{x}) = E^{w}_{Y|X}\left[\frac{y}{C_{X,Y}(x,\mathrm{y})}|x\right]$$

where

$$w(x,y) = C_{X,Y}(x,\mathrm{y})^2 e^{-yC_{X,Y}(x,\mathrm{y})\hat{f}^{(m)}(\mathrm{x})}$$

$$\min_{G(\mathrm{x})} E^{w}_{Y|X}\left[\left(\frac{y}{C_{X,Y}(x,\mathrm{y})} - G(\mathrm{x})\right)^2\right],$$

Which is equivalent to weighted least square regression of $y_i$ to $x_i$ using weights $w_i$ as given by () and () the optimum feature is one of the smallest regression errors.